Flash Storage and IO Operation

Lecture Objectives:

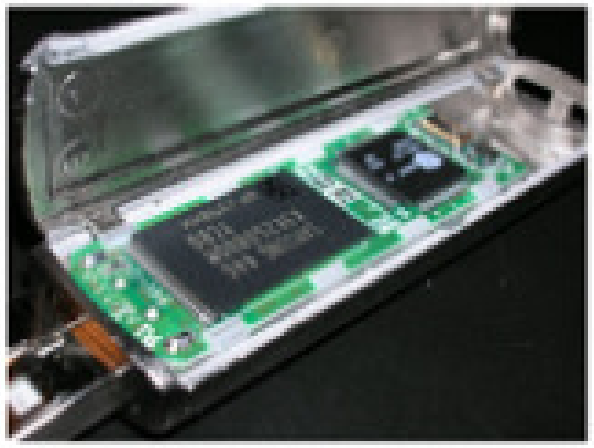
1) Explain the limitations of flash memory.
2) Define wear leveling.
3) Define the term IO Transaction
4) Define the terms synchronous bus and asynchronous interconnect.
5) Explain the difference between polling and interrupts

Flash Storage

- Nonvolatile semiconductor storage
  - 100× – 1000× faster than disk
  - Smaller, lower power, more robust
  - But more $/GB (between disk and DRAM)

Better performance

More expensive



MSOE

Flash Memory Performance

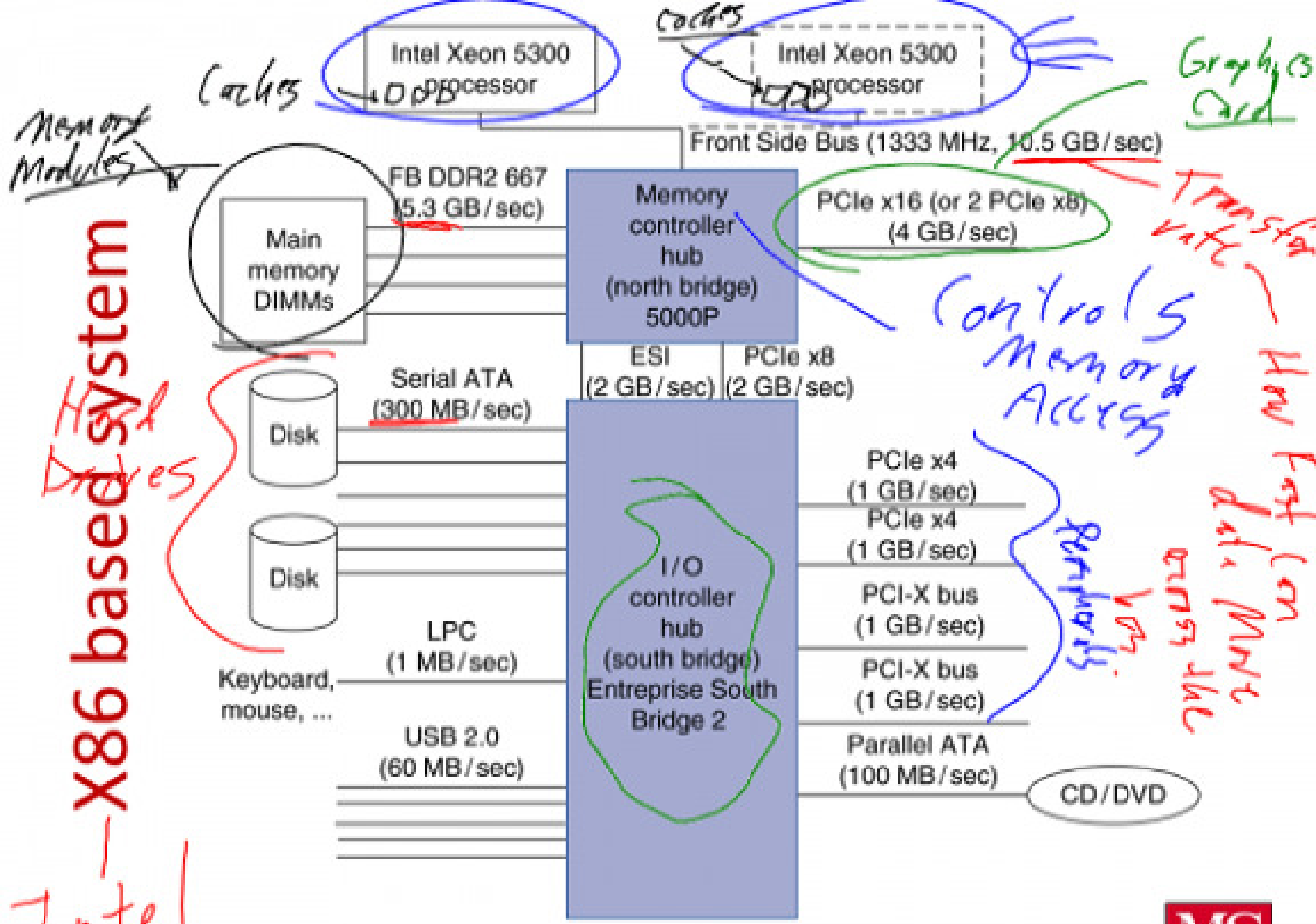
Characteristics	NOR Flash Memory	NAND Flash Memory
Typical use	BIOS memory	USB key
Minimum access size (bytes)	512 bytes	2048 bytes
Read time (microseconds)	0.08	25
Write time (microseconds)	10.00	1500 to erase + 250
Read bandwidth (MBytes/second)	10	40
Write bandwidth (MBytes/second)	0.4	8
Wearout (writes per cell)	100,000	10,000 to 100,000
Best price/GB (2008)	$65	$4

- Limitations
  - Cost is higher than mechanical drives
  - Cells wear out over time


Read/
write
a
status
DDDDDDDD
BitST
2048X8 bits
ERASE
EVERYthing
FFFFFF
FF....
All ones

Definition

- Wear leveling
  - A technique which organizes blocks of data so that blocks which have been rewritten many times are exchanged for blocks which have not been written as often.

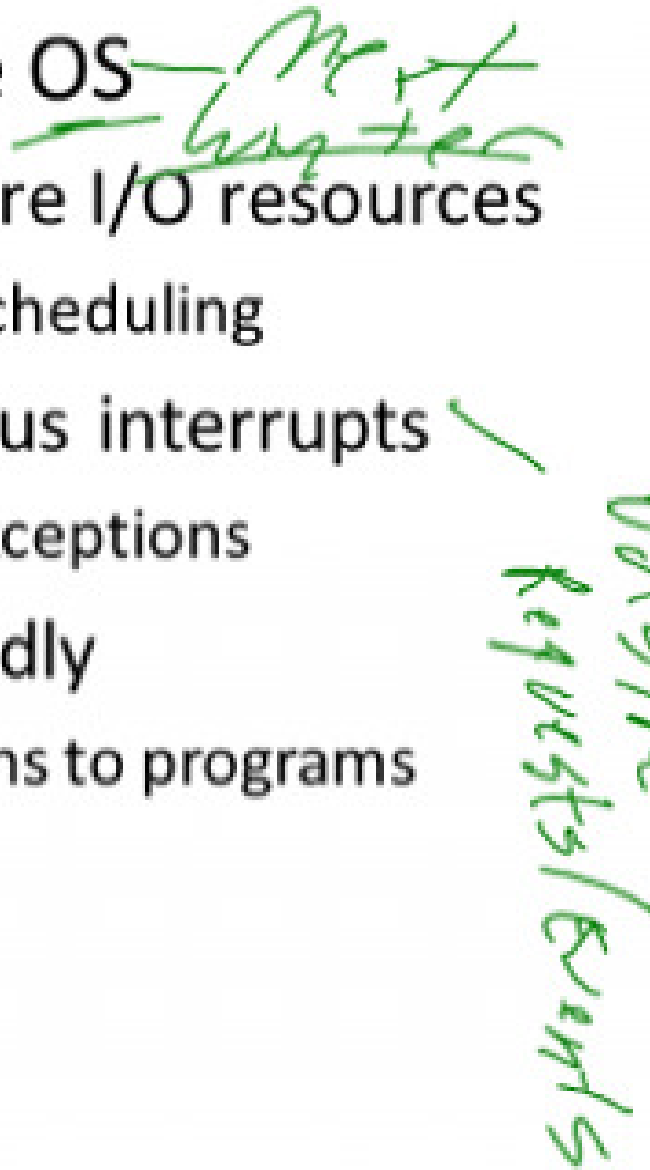

Intel Xeon 5300 processor
Intel Xeon 5300 processor
Front Side Bus (1333 MHz, 10.5 GB/sec)
Main memory DIMMs
FB DDR2 667 (5.3 GB/sec)
Memory controller hub (north bridge) 5000P
PCIe x16 (or 2 PCIe x8) (4 GB/sec)
ESI (2 GB/sec)
PCIe x8 (2 GB/sec)
Serial ATA (300 MB/sec)
Disk
Disk
I/O controller hub (south bridge) Entreprise South Bridge 2
PCIe x4 (1 GB/sec)
PCIe x4 (1 GB/sec)
PCI-X bus (1 GB/sec)
PCI-X bus (1 GB/sec)
Parallel ATA (100 MB/sec)
CD/DVD
LPC (1 MB/sec)
Keyboard, mouse, ...
USB 2.0 (60 MB/sec)

Definitions

- Processor memory bus
  - A bus that connect the processor to the memory, and is generally high speed.
- Backplane bus
  - A bus that is designed to allow processors, memory, and I/O devices to coexist on a single bus
- IO Transaction
  - A sequence of operations over the interconnect that includes a request, and may include a response either of which may carry data.
- Synchronous bus
  - A bus that includes clock and control lines and a fixed protocol for communicating that is relative to the clock
- Asynchronous interconnect
  - A mechanism which uses a handshaking protocol rather than a clock to accommodate devices of varying speeds.

I/O Management

- I/O is mediated by the OS
  - Multiple programs share I/O resources
    - Need protection and scheduling
  - I/O causes asynchronous interrupts
    - Same mechanism as exceptions
  - I/O programming is fiddly
    - OS provides abstractions to programs



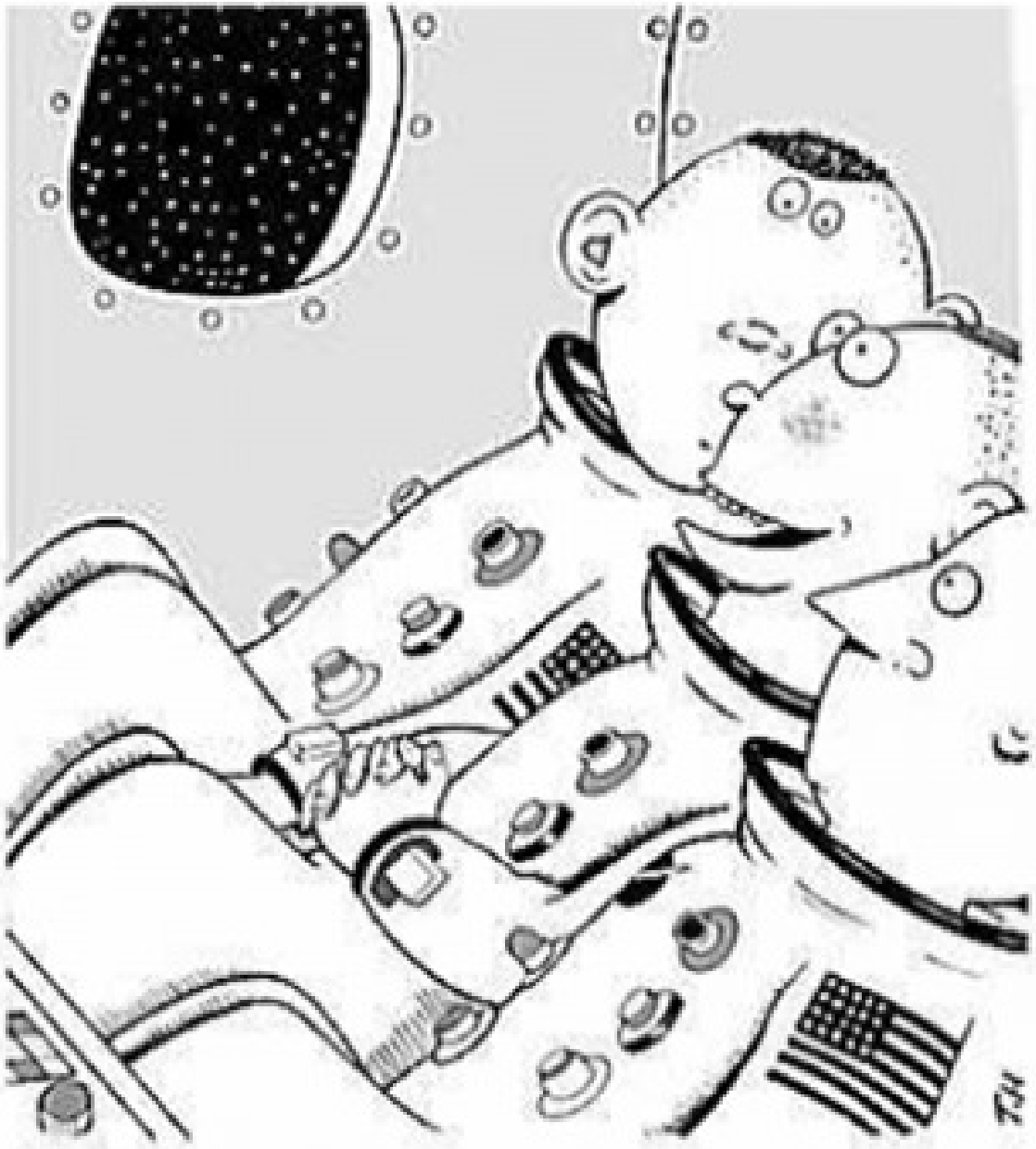
I/O Commands

- I/O devices are managed by I/O controller hardware
  - Transfers data to/from device
  - Synchronizes operations with software
- Command registers
  - Cause device to do something
- Status registers
  - Indicate what the device is doing and occurrence of errors
- Data registers
  - Write: transfer data to a device
  - Read: transfer data from a device

I/O Register Mapping

- Memory mapped I/O — I/O is really just an address.
  - Registers are addressed in same space as memory
  - Address decoder distinguishes between them
  - OS uses address translation mechanism to make them only accessible to kernel
- I/O instructions
  - Separate instructions to access I/O registers
  - Can only be executed in kernel mode
  - Example: x86

Polling



"Are we there yet? Are we there yet? Are we there yet?"

Polling

- Periodically check I/O status register
  - If device ready, do operation
  - If error, take action
- Common in small or low-performance real-time embedded systems
  - Predictable timing
  - Low hardware cost
- In other systems, wastes CPU time

Interrupts

No! Shut up and I'll tell you when we are there!

Interrupts

MSOE

Interrupts

- When a device is ready or error occurs
  - Controller interrupts CPU
- Interrupt is like an exception
  - But not synchronized to instruction execution
  - Can invoke handler between instructions
  - Cause information often identifies the interrupting device
- Priority interrupts
  - Devices needing more urgent attention get higher priority
  - Can interrupt handler for a lower priority interrupt